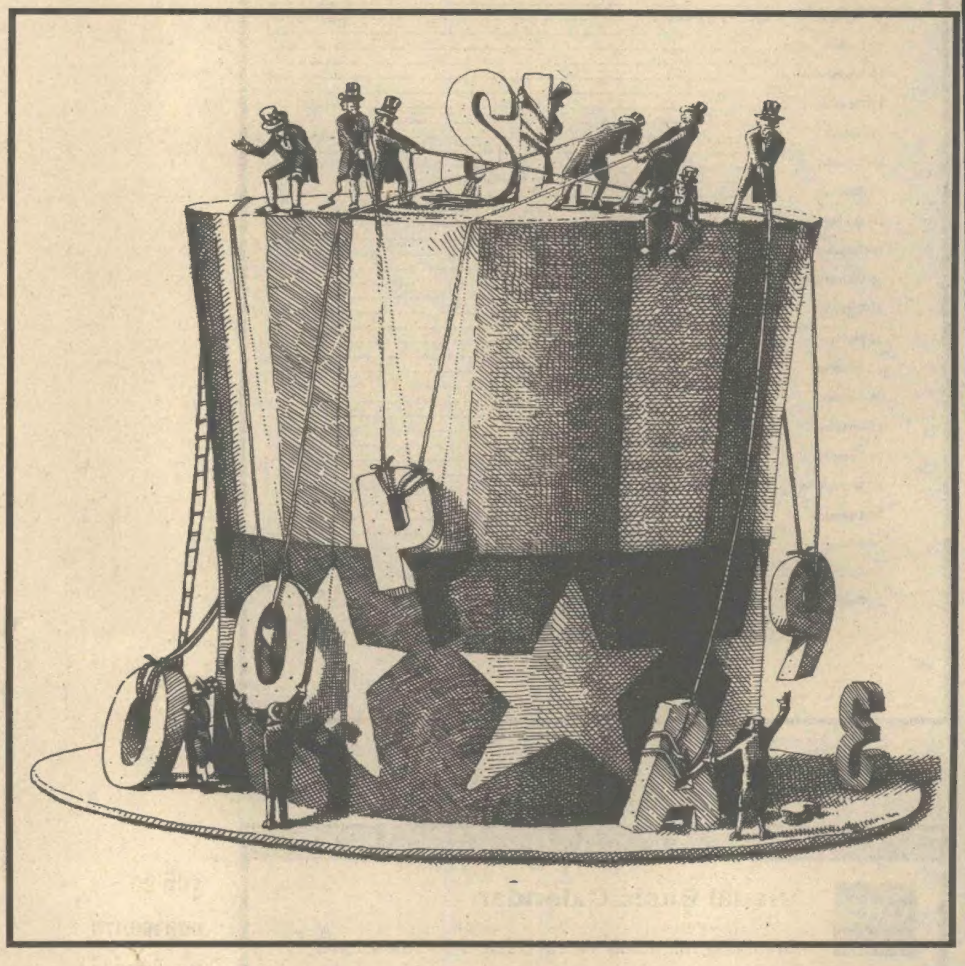
บุญเลิศ เอี่ยมทัศนา

เงาโอโอพี
ในวิชวลเบสิก

นำเสนอเรื่องราวของวิชวลเบสิกกับโอโอพี และแนะนำผู้ที่เขียนโปรแกรมบนวินโดวส์ได้ทดลองใช้วิชวลเบสิก



ผมได้ยินได้ฟังข่าวของวิชวลเบสิกมานานแต่ยังไม่สบโอกาส จนมีเหตุต้องนำมาใช้จึงมีเรื่องมาเล่าสู่กันฟัง เป็นเรื่องของวิชวลเบสิกกับโอโอพี และเมื่อเขียนแล้วเลยเพิ่มเนื้อหาให้ผู้ที่เขียนโปรแกรมบนวินโดวส์ด้วยภาษาอื่นได้ทดลองใช้วิชวลเบสิก รู้หลายภาษาดีกว่าภาษาเดียว แต่คงไม่หลงเสน่ห์สาววิชวลเบสิกจนลืมเพื่อนเก่า

ความเป็นมา

มีผู้ถามผมว่าทำไมผมถึงแนะนำการเขียนโปรแกรมแบบโอโอพี (OOP : Object-Oriented Programming) ผมตอบว่าการเขียนโปรแกรมตามแบบ (ดั้งเดิมที่เคยเขียนกันมา) กำลังจะหมดไป แต่ก็ไม่หายไปไหนไปอยู่ในโอโอพี

เขาแย้งว่าวิชวลเบสิก (Microsoft Visual Basic) ซึ่งเป็นคอมพายเลอร์รุ่นใหม่ ทำไมจึงไม่เขียนแบบโอโอพีและทำไมจึงไม่ให้เขียนโปรแกรมในแบบโอโอพี

ผมฟังแล้วไม่ตอบว่าอะไร พอมีโอกาสก็จัดหาวิชวลเบสิกมา เมื่อทดลองเขียนโปรแกรมก็ขำครับ เพราะกำลังเขียนโปรแกรมแบบโอโอพีอยู่ เป็นโอโอพีที่ถูกซ่อน คนอเมริกันคงกลัวโอโอพีเหมือนกัน

เหมือนคนที่ตั้งปัญหาถามผม

ที่ว่าเป็นโอโอพี

เมื่อเห็นว่าเป็นโอโอพีและน่าจะเป็นโอโอพีเป็นอย่างยิ่ง ผมจึงนำหลักการของโอโอพีเข้าจับ โดยที่โปรแกรมในแบบโอโอพีจะต้องมีคุณสมบัติดังนี้ คือ

1. Encapsulation ข้อมูลและกิจกรรมจะต้องรวมอยู่ในที่แห่งเดียวกันและอยู่ในปกป้อง ดังที่ได้สร้างเป็นสกุล

2. Inheritance ให้สืบสกุลได้

3. Polymorphism ให้แตกแขนงสกุลได้ (หาคำนี้มานานแล้วครับ)

ดังเช่นในเทอร์โบปาสคาลได้สร้างสกุลสืบสกุลมาเป็นทอด ๆ จนถึง TWindow เป็นสกุลของหน้าต่างแล้วแตกแขนงออกไปหลายสาขา สาขาหนึ่งคือ TControl ซึ่งแตกแขนงเป็นสกุลคอนโทรลต่าง ๆ ทำงานต่าง ๆ กัน มีที่เหมือนกันคือความเป็นหน้าต่าง

บทความของผมในฉบับนี้จึงจะกล่าวถึงเรื่องนี้ครับ แต่จะมีเรื่องอื่นเข้ามาแทรกเพื่อแสดงถึงข้อแตกต่างกับภาษาอื่น และการเขียนโปรแกรมด้วยวิชวลเบสิกด้วย

แยกพวก

จนถึงปัจจุบันมีคอมพายเลอร์แต่ขอนับเป็นภาษา ที่ให้เราเขียนโปรแกรมบนวินโดวส์ได้อยู่ 4 ภาษาคือ C, C++ (ของหลายบริษัท) เทอร์โบปาสคาล (รวมทั้งบอร์แลนด์ปาสคาลด้วย) และวิชวลเบสิก ซึ่งผมขอแยกพวกแยกกลุ่มดังนี้

- สามภาษาแรกเป็นภาษาหลักที่ใช้วิธีเตรียมโปรแกรม (เขียน ป้อน และคอมพายล์) ในแนวทางเดียวกัน

โดยที่ C เป็นภาษาตามแบบ ส่วน C++ และเทอร์โบปาสคาลเป็นโอโอแอล (OOL : Object Oriented Language)

- ส่วนวิชวลเบสิกขอถือว่าเป็นภาษาสำเร็จรูปไว้ก่อน

สำเร็จรูปอย่างไร ก็เหมือนของสำเร็จรูปทั้งหลาย (ทั้งด้านดีและไม่ดี)

เทียบภาษา

แต่เมื่อเป็นคอมพายเลอร์ผมจึงเทียบกับหม้อหุงข้าว โดยให้วิชวลเบสิกเป็นหม้อหุงข้าวไฟฟ้าอัตโนมัติ นอกนั้นเป็นหม้อหุงข้าวธรรมดา จึงต้องเทียบกันตรงเตา ซึ่งผมจะเทียบตรงขั้นตอนของการเตรียมโปรแกรม (เขียน ป้อน และคอมพายล์) จะได้ว่า

- ของวิชวลเบสิกเรียกได้ว่าพอเปิดปุ๊บก็ติดปั๊บ

- ของเทอร์โบปาสคาลเทียบได้กับเตาแก๊ส พอเปิดก็หุงข้าว (รันโปรแกรม) ได้ทันที แต่ก็ต้องปรับแต่งเปลวไฟ

- ของ C++ เทียบได้กับเตาถ่าน ต้องพัดไฟกันก่อน

- ของ C ใช้ฟืนครับ ต้องเป่ากันจนตาแดง กว่าจะได้ป้อนโปรแกรม

ส่วนของใครจะสุกเร็วกว่ากันคือรันเร็วกว่ากัน เทียบไม่ได้ครับในเครื่องรุ่นใหม่ และต่างก็เรียกใช้ฟังก์ชันของวินโดวส์ด้วยกันทั้งนั้น

ขอติ

ของทุกอย่างย่อมมีทั้งข้อดีและเสีย ของวิชวลเบสิกผมเห็นข้อติอยู่ข้อครึ่ง ข้อแรกต้องขอใช้คำไทยปนอังกฤษคือ ไม่ portable กล่าวคือ เมื่อผมเขียนโปรแกรมด้วยภาษาหลักคอมพายล์เป็นไฟล์ .EXE แล้ว ผมสามารถนำไปรันกับวินโดวส์บนเครื่องใดก็ได้

แต่ของวิชวลเบสิกไม่ได้ครับ ต้องรันกับเครื่องที่ติดตั้งวิชวลเบสิกรุ่นเดียวกัน หรือมีไฟล์ vbrunxxx.dll (xxx คือรุ่น)

ทำไมถึงเป็นเช่นนี้ผมคิดเอาเองว่าเป็นเพราะ ไมโครซอฟต์คงต้องการให้ไฟล์ .exe เล็ก และคงเห็นว่า ในบรรดาผู้เขียนโปรแกรมคือผู้ที่ซื้อซอฟต์แวร์ไปใช้ เช่น 100 คน จะมีผู้เขียนโปรแกรมจริงจังที่เรียกกันว่ามืออาชีพอยู่เพียง 5-10 คนเท่านั้น ซึ่งไมโครซอฟต์ก็มี Visual C++ อยู่แล้ว นอกจากนั้นคือผู้ที่อยู่ตามบ้านตามสำนักงาน เขียนโปรแกรมอยู่ในวงแคบ คือกลุ่มเป้าหมายครับ

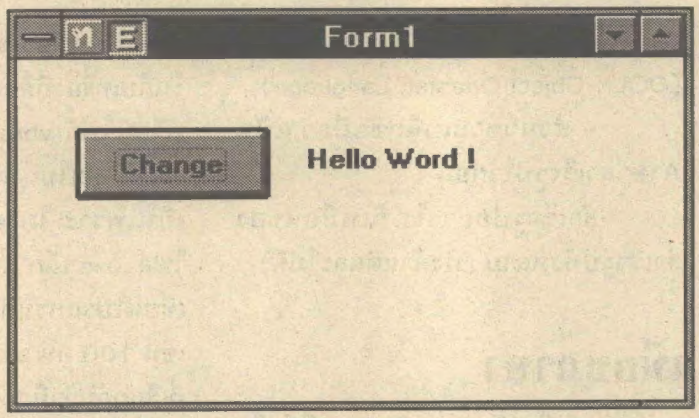
อีกครึ่งข้อก็คือความสำเร็จรูป (กับมีเรื่องที่ผมจะบ่นต่อไปอีก) ผมเกรงไปว่าผู้ที่รู้จักแต่วิธีหุงข้าวด้วยหม้อไฟฟ้าจะหุงข้าวด้วยหม้อธรรมดาไม่ได้

ดูที่โปรแกรม

เมื่อจะดูว่าเป็นโอโอพีหรือไม่ก็ต้องดูกันที่โปรแกรม แต่ของวิชวลเบสิกจะต้องดูการเตรียมโปรแกรมทั้งหมดเพราะเกี่ยวพันกัน ผมจึงขอทบทวนทั้งหมดครับ คือ

ผมจะเขียนโปรแกรมแสดงภาพหน้าต่างดังรูปที่ 1 เมื่อผู้ใช้เลือกปุ่ม OK จะเปลี่ยนข้อความ Hello-Word! เป็น Wellcome to Visual Basic ซึ่งถ้าผมเขียนโปรแกรมด้วยเทอร์โบปาสคาลจะเขียนโปรแกรมได้ดังรูปที่ 2 (แสดงเฉพาะส่วนที่ต้องการเทียบกันเท่านั้น) ถ้าผมจะเขียนด้วยวิชวลเบสิกจะต้องดำเนินการดังนี้ครับ

1. เปิดโปรเจ็กต์ใหม่โดยเลือกรายการ FileNew Project จะแสดงภาพดังรูปที่ 3 หน้าต่างที่มีคำว่า Form1 ก็คือหน้าต่างแต่ในวิชวลเบสิกเรียกว่าฟอร์ม (Form)



รูปที่ 1
ผลการรันโปรแกรมแรก

```
TYPE
  CommandCP = ^CommandC;
  CommandC  = OBJECT(TButton)
    oInfoP  : PStatic;
 constructor Init(apar : PWindowsObject; oObjP : PStatic);
 procedure  wmLButtonDown(var msg : TMessage);
              virtual wm_First + wm_LButtonDown;
              END;


  WndProCP = ^WndProC;
  WndProC = OBJECT(TWindow)
 constructor  Init(apar : PWindowsObject; atitle : PChar);
            END;

(* ------------------- Implementation Part  ----------------- *)


CONSTRUCTOR  CommandC.Init(apar : PWindowsObject; oObjP : PStatic);
BEGIN
  Tbutton.Init(apar, 101, 'Change', 50,60,100,50, true);
  oInfoP := oObjP;
END;


PROCEDURE  CommandC.wmLButtonDown(var msg : TMessage);
BEGIN
  oInfoP^.SetText('Wellcome to Visual Basic.');
END;


CONSTRUCTOR  WndProC.Init(apar : PWindowsObject; atitle : PChar);
VAR  oLabelP : PStatic; oButtP : CommandCP;
BEGIN
  TWindow.Init(apar,atitle);
  oLabelP := New(PStatic, Init(@self, -1,
                        'Hello World !',200, 70, 150, 30, 30));
  oButtP := New(CommandCP, Init(@self, oLabelP));
END;
```

รูปที่ 2 ส่วนของโปรแกรมในภาษาปาสคาล

2. ผมต้องการแสดงข้อความในกรอบ Static control แต่ในวิชวลเบสิกเรียกเป็น Label Control (ทำไมต้องเปลี่ยนชื่อไม่ทราบ) จะต้องเลือกเครื่องมือนี้ในกรอบเครื่องมือทางซ้าย คือที่เป็นปุ่มอักษร A แล้ว

- นำไปกำหนดขนาดและตำแหน่ง (คลิ๊กเม้าส์ที่มุมบนซ้ายแล้วเลื่อนไปปล่อยมือที่มุมล่างขวา) ดังที่เห็นในรูปที่ 3

- ที่หน้าต่าง Properties เลือกหัวข้อ Caption แล้วป้อนแก้ข้อความที่กรอบป้อน (อยู่ทางขวาของเครื่องหมายถูก) จาก Label1 เป็น Hellow World!

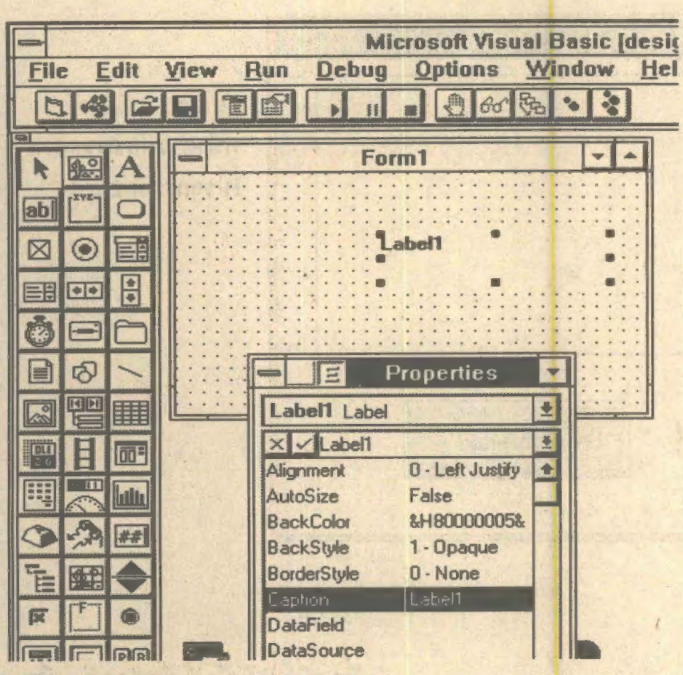
3. ดำเนินการเช่นเดียวกันนี้ แต่เลือกเครื่องมือเป็น Command control (เปลี่ยนชื่ออีกเช่นกัน) แล้วป้อนแก้ข้อความที่ Caption เป็น Change

4. ป้อนโปรแกรม ถึงขั้นตอนป้อนโปรแกรมครับ ขอให้คลิ๊กซ้อน (double chick) ที่ปุ่ม Change จะแสดงหน้าต่าง Code Window ดังรูปที่ 4 ให้

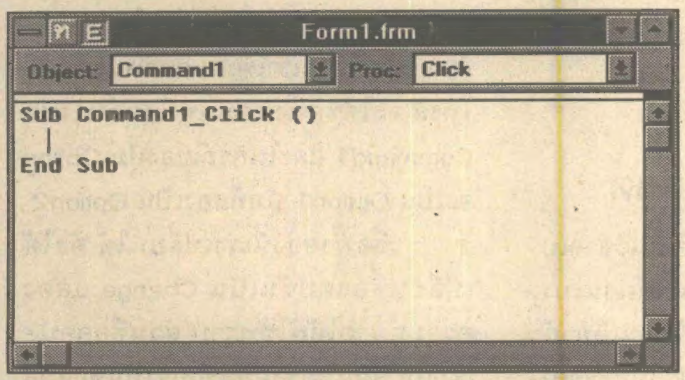
- เลือกโพรซีเดอร์ในกรอบ Proc ที่ต้องการคือ Click จึงไม่ต้องเลือก (คำว่า Procedure นี่ก็เหมือนกัน ควรใช้เป็น Method เพราะมีทั้งโพรซีเดอร์และฟังก์ชัน)

- ป้อนโปรแกรมตามลิสติงที่ 1 ซึ่งป้อนเพียงบรรทัดเดียว ผมเรียกเป็นลิสติงเพราะเป็นโปรแกรมที่สมบูรณ์แล้วสำหรับแอพพลิเคชันนี้ ซึ่งจะอยู่ในไฟล์ .frm (จะรวมการดำเนินการในข้อ 2 เก็บไว้ในไฟล์นี้ด้วย)

เท่านี้ครับง่ายดี คุณสามารถรันโปรแกรมได้เลยโดยกดคีย์ F5 จะได้ผลการรันตามที่กล่าวมา



รูปที่ 3
การกำหนดคอนโทรลในฟอร์ม ทางด้านซ้ายเป็นกล่องเครื่องมือ มีหน้าต่าง Properties อยู่ด้านล่าง แสดงไฮไลต์ตรงหัวข้อ Caption



รูปที่ 4
หน้าต่าง Code Window สำหรับป้อนโปรแกรม

```
Sub Command1_Click ()
  Label1.Caption = "Well come to Visual Basic."
End Sub
```

ลิสติงที่ 1

ข้อแตกต่าง

จากโปรแกรมนี้จะเห็นข้อแตกต่างข้อหนึ่ง ซึ่งจะพาไปสู่ข้อแตกต่างอีกหลายข้อ กล่าวคือ เมื่อเราเขียนโปรแกรมด้วยภาษามาตรฐาน จะใช้คอนโทรลในไดอะล็อกบ็อกซ์อย่างเป็นของธรรมดา ไม่ค่อยใช้ในหน้าต่าง หนังสือภาษาซีส่วนมากไม่เอ่ยถึง มีในหนังสือที่ใช้เทอร์โบปาสคาล (ของผมก็มี)

และเมื่อใช้ไม่ว่าในหน้าต่างหรือในไดอะล็อกบ็อกซ์ จะไม่ค่อยเขียนโปรแกรมให้แก่คอนโทรล หน้าต่างหรือไดอะล็อกบ็อกซ์จะเป็นผู้ควบคุมการทำงานของคอนโทรล ด้วยวิธีรับ-ส่งเมสเสจกับคอนโทรล ไม่ได้ให้คอนโทรลติดต่อกันเอง ตามโปรแกรมในรูปที่ 2 เป็นเพราะผมเจตนาจะเขียนเพื่อเปรียบเทียบ ไม่ใช่วิธีเขียนโปรแกรมตามปกติ

แต่ในวิชวลเบสิกต้องขอเรียกว่ากลับตาลปัตรครับ คือ ใช้คอนโทรลในหน้าต่างอย่างเป็นธรรมดา ไม่รับ-ส่งเมสเสจกับคอนโทรล เขียนโปรแกรมแก่คอนโทรล และให้คอนโทรลติดต่อกันเอง

สกุลอยู่ที่ไหน

เมื่อเป็นโอโอพีก็ต้องมีสกุล หากเทียบกับเทอร์โบปาสคาล Label ก็คือสกุล TStatic แล้วสกุล Label อยู่ตรงไหน? ถ้าถามไมโครซอฟต์คงตอบว่าไม่มี (เพราะกลัวโอโอพี) แล้วอยู่ไหน อยู่ตรงหน้าครับ

เช่นเดียวกับของเทอร์โบปาสคาลที่เราไม่เคยเห็นโปรแกรมของ TStatic แต่ทราบว่ามีและรู้จักฟิลด์ข้อมูลซึ่งทางด้านโอโอพีเรียกว่า Property และรู้จักกิจกรรมซึ่งทางด้านโอโอพีเรียกว่า Behavior หรือ Method (คือที่ผมใช้คำว่ากิจกรรม) กิจกรรมก็คือโปรแกรมย่อย โปรแกรมย่อยก็คือ Procedure

ตรงหน้าเรามีทั้ง Property (ในหน้าต่าง Properties ดังในรูปที่ 3) และ Procedure (ในหน้าต่าง Code Window ดังในรูปที่ 4 แต่มีชื่อหน้าต่างเป็น Form1 .frm) จึงขอสมมติว่ามีสกุลไว้ก่อนครับ

สืบสกุลอย่างไร

เมื่อมีสกุลแล้วหากต้องการให้เป็นโอโอพียิ่งขึ้นจะต้องมีการสืบสกุล และขอรวมการแตกแขนงไว้ด้วยกัน เพราะการสืบสกุลเทียบได้กับบอกว่าให้มีลูกได้ ส่วนการแตกแขนงคือให้มีได้หลายคน

ผู้สร้างวิชวลเบสิกฉลาดมากครับ ขอให้เทียบโปรแกรมของกิจกรรม CommandC.wmLButtonDOWN ในรูปที่ 2 กับ Command1_Click ในลิสติงที่ 1 จะเห็น

ว่าต่างก็เป็นโปรแกรมของกิจกรรม มีส่วนหัวประกอบด้วยชื่อสกุลซึ่งของวิชวลเบสิกและความจริงในเทอร์โบปาสคาลก็เรียกเป็นออปเจ็กต์ (Object) กับชื่อกิจกรรมของเทอร์โบปาสคาลแยกจากกันด้วยจุด ของวิชวลเบสิกแยกด้วยขีดล่าง (under score) ของ C++ แยกด้วยเครื่องหมาย ::

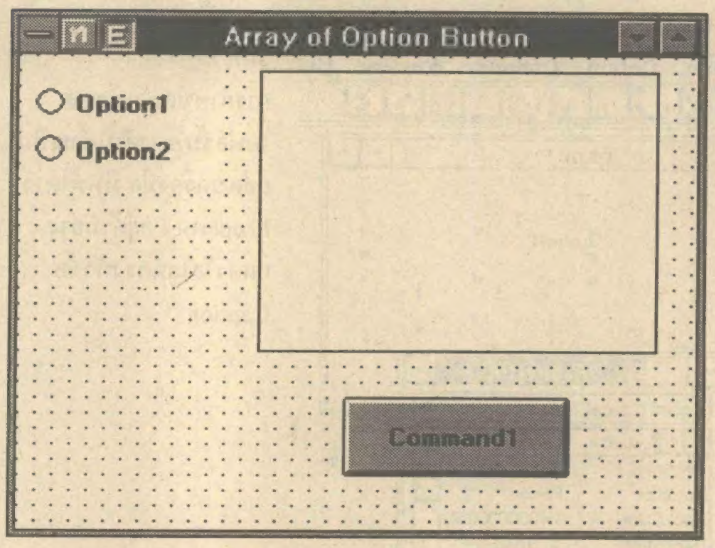
นั่นคือโปรแกรมที่เราป้อนเป็นโปรแกรมของกิจกรรม ซึ่งจะป้อนได้ในสกุลใหม่เท่านั้น แต่เราไม่ได้สร้างสกุลใหม่ดังที่ผมต้องสร้างสกุล CommandC ให้แก่ปุ่มนี้ในเทอร์โบปาสคาล ปัญหาคือทำได้อย่างไร

ผมคิดเอาเองว่าในวิชวลเบสิกจะต้องมีสกุลของคอนโทรลทุกชนิด ดังเช่นปุ่ม Command อาจมีสกุลและให้ชื่อว่า TCommand เช่นเดียวกับที่ในเทอร์โบปาสคาลมี TButton แล้วสร้างต้นแบบ (template) ของผู้สืบสกุลไว้ เช่น ให้ชื่อว่า TCommandTpl มีกิจกรรมครบถ้วน แต่เป็นโปรแกรมเปล่าดังในรูปที่ 4

เมื่อเรากำหนดคอนโทรล ซึ่งในวิชวลเบสิกถือว่าเป็นการกำหนดออปเจ็กต์ จะลอกต้นแบบให้แล้วนำชื่อออปเจ็กต์เช่น Command1 ไปเป็นชื่อสกุล (ซึ่งการเขียนโปรแกรมในระดับคอมพายเลอร์ทำได้) เช่นเดียวกันหากเรากำหนดปุ่มชนิดนี้อีกก็จะได้ชื่อสกุลเป็น Command2

นั่นคือจะมีสกุล 1 สกุลต่อ 1 ออปเจ็กต์ ซึ่งทางด้านโอโอพีสกุลหนึ่ง ๆ ให้กำหนดได้หลายออปเจ็กต์ และข้อความที่ผมกล่าวมานี้หมายถึงหน้าต่าง ซึ่งในวิชวลเบสิกเรียกว่าฟอร์ม (Form) ด้วย

เมื่อเป็นเช่นนี้จะไม่ถือว่าเป็นการหมดเปลืองโดยใช่เหตุหรือ? ไม่หรอกครับ เพราะในโอโอแอลการเรียกใช้กิจกรรมจะเรียกผ่านตาราง ต้นแบบที่ลอกให้จึงเป็นเพียงตาราง (ถึงจะหมดเปลืองก็ไม่มาก) และก็เป็นธรรมดาของการเขียนโปรแกรมบนวินโดวส์อยู่แล้ว ที่ใช้สกุล 1 สกุลต่อ 1 ออปเจ็กต์ เพราะส่วนมากจะเป็นโปรแกรมของหน้าต่างซึ่งทำงานแตกต่างกัน



รูปที่ 5
หน้าต่างฟอร์มของโปรแกรมที่ 2

อะเรย์ของออปเจ็กต์

แต่ถ้าไม่ธรรมดาก็คือในวิชวลเบสิกที่ต้องเขียนโปรแกรมให้แก่คอนโทรล ซึ่งคอนโทรลส่วนมากจะทำงานเดียวกัน แตกต่างกันตรงค่าในตัวแปร ในวิชวลเบสิกจึงให้กำหนดอะเรย์ของออปเจ็กต์ของสกุลเดียวกันได้ ซึ่งก็เป็นธรรมดาเพราะถ้าเราให้กำเนิดออปเจ็กต์เป็นจำนวนมากก็จะให้อยู่ในอะเรย์

ดูตัวอย่างของจริง ตามรูปที่ 5 เป็นฟอร์มของโปรแกรมที่จะกล่าวต่อไป ในระหว่างเตรียมโปรแกรม ซึ่งจะดำเนินการได้โดยกำหนดคอนโทรลดังนี้

1. PictureBox ด้วยเครื่องมือทางซ้ายของปุ่ม Label (ที่กล่าวถึงในตัวอย่างแรก) ที่ตีไว้เป็นกรอบใหญ่
2. ปุ่ม Command1
3. ปุ่มวิทยุ (radio button) ซึ่งในวิชวลเบสิกเรียกเป็น OptionButton 2 ปุ่ม

มีเรื่องที่ผมยังไม่ได้กล่าวคือ ชื่อออปเจ็กต์ (Name) ไม่ใช่ชื่อแคปชัน (Caption) หากเราเพียงแค่กำหนดคอนโทรล จะได้ชื่อทั้งสองนี้เหมือนกัน เช่น Command1 และในกรณีของปุ่ม Option จะเป็น Option1 ปุ่มที่สองเป็น Option2

ชื่อทั้งสองนี้เราเปลี่ยนได้ ดังได้เปลี่ยนชื่อแคปชันเป็น Change แสดงความหมายให้ผู้ใช้ทราบ ส่วนชื่อออปเจ็กต์เปลี่ยนได้เช่นกันเพื่อใช้เป็นชื่อสกุล เป็นการตั้งชื่อเพื่อประโยชน์ในการเขียนโปรแกรมครับ และในการนี้ต้องตั้งชื่อให้ด้วยเหตุผลที่จะกล่าวต่อไป ขอให้ดำเนินการดังนี้

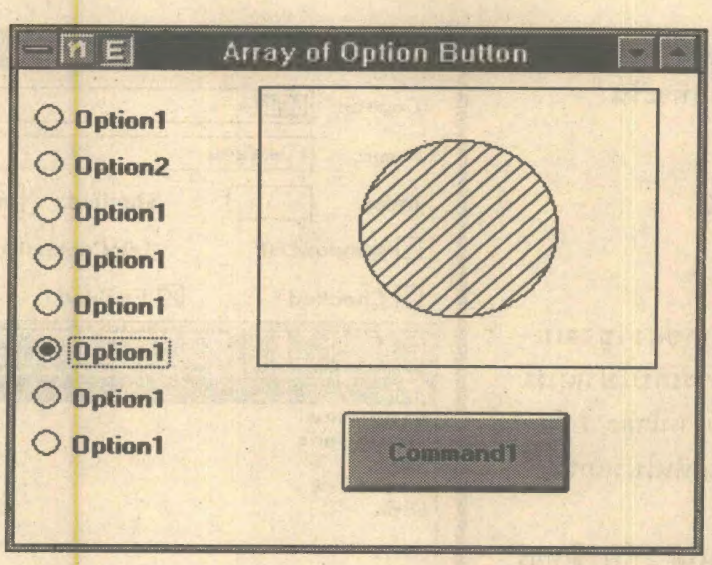
1. เลือกปุ่มแรกคือ Option1 แล้วเลือกรายการ Name แก้ชื่อจาก Option1 เป็น optButton
2. เลือกปุ่มที่สองคือ Option2 เปลี่ยนชื่อออปเจ็กต์เป็น optButton เช่นกัน เพราะชื่อซ้ำกันวิชวลเบสิกจะถามว่าต้องการให้กำเนิด control array หรือให้ตอบ Yes และตามรูปที่ 5 ได้เปลี่ยนชื่อแคปชันเป็น Option2

การดำเนินการตามข้อ 2 นี้เป็น

การให้กำเนิด (create) อะเรย์ของออปเจ็กต์ มีชื่ออะเรย์เป็น optButton และเป็นเหตุผลสนับสนุนอีกข้อหนึ่งว่าชื่อออปเจ็กต์คือชื่อสกุล

หากรันโปรแกรมนี้ (การป้อนโปรแกรมจะกล่าวในหัวข้อต่อไป) แล้วเลือกปุ่ม Command1 รวม 6 ครั้ง และเลือกปุ่ม Option (ปุ่มที่เห็นว่าได้รับโฟกัสอยู่ในขณะนี้) จะได้ผลการรันดังรูปที่ 6

นั่นคือในโปรแกรมจะให้กำเนิดปุ่มออปชันอีก 6 ปุ่ม เมื่อเลือกปุ่มใดจะแสดงวงกลมในกรอบภาพมีแพตเทอร์นเปลี่ยนไปตามแต่ละปุ่ม



รูปที่ 6
ผลการรันของโปรแกรมที่ 2

องดีใน Help

ผมมีคู่มือของไมโครซอฟต์เกือบครบชุด แต่ระยะหลังนี้แทบไม่ได้ใช้ ใช้ Help ของเทอร์โบปาสคาลแทนก็ติดใจมากครับ แต่มาถึง Help ของวิชวลเบสิกต้องเรียกว่าโปรดปราน เพราะมีตัวอย่างในการเขียนโปรแกรมกับวินโดวส์ด้วย ตัวอย่างก็เหมือนภาพครับ หนึ่งตัวอย่างดีกว่าคำอธิบายพันคำ

และยิ่งไปกว่านั้น ส่วนมากเป็นตัวอย่างที่เป็นโปรแกรมที่สมบูรณ์สามารถลอกไปรันได้เลยครับ ผมก็ขอท้าวความว่าผมได้ตัวอย่างนี้มาได้อย่างไร ผมได้วิชวลเบสิก 3.0 มาพร้อมกับหนังสือเล่มหนึ่ง

พอดูตัวอย่างแรกแล้วก็อยากเขียนโปรแกรมเอง ตามที่เขียนโปรแกรมแรกต้องการเพิ่มปุ่ม Command อีกปุ่มหนึ่ง ตามความเคยชินผมให้ชื่อออปเจ็กต์ (Name) ซึ่งผมถือว่าเป็นชื่อสกุลให้เป็นชื่อเดียวกัน ก็ถูกท้วงว่าจะให้เป็น control array หรือ ก็เปิดดูคำนี้ใน Help และก็ไล่ไปตามลำดับดังนี้ครับ

1. Control arrays
2. Creating a Control Array
3. See Also
4. Index Property (Control Array)
5. Example

จะแสดงตัวอย่างพร้อมด้วยคำอธิบายให้ดำเนินการก่อนคือ การกำหนดคอนโทรลต่าง ๆ ตามที่ผมกล่าวมา ส่วนตัวอย่างตามลิสติงที่ 2 ขอให้

1. ลอก (Copy) ตัวอย่างลงคลิปบอร์ด
2. กลับไปที่ฟอร์ม คลิ๊กซ้อนจะแสดงหน้าต่าง Code Window (มีวิธีอื่นอีก)
3. ที่กรอบ Object ให้เลือกรายการเป็น (general)
4. ปะ (paste) ข้อความจากคลิปบอร์ด

ลิสติงที่ 2

```
Sub Command1_Click ()
    Static MaxIdx    ' Largest index in array.
    If MaxIdx = 0 Then MaxIdx = 1    ' Preset MaxIdx.
    MaxIdx = MaxIdx + 1 ' Increment index.
    If MaxIdx > 7 Then Exit Sub ' Put 8 buttons on form.
    Load OptButton(MaxIdx)  ' Create new item in array.
    ' Set location of new option button under previous button.
    OptButton(MaxIdx).Top = OptButton(MaxIdx - 1).Top + 360
    OptButton(MaxIdx).Visible = True     ' Make new button visible.
End Sub

Sub OptButton_Click (Index As Integer)
    Dim H, W     ' Declare variables.
    Picture1.Cls     ' Clear picture.
    Picture1.FillStyle = Index  ' Set FillStyle.
    W = Picture1.ScaleWidth / 2 ' Get size of circle.
    H = Picture1.ScaleHeight / 2
    Picture1.Circle (W, H), W / 2    ' Draw circle.
End Sub
```

วิชวลเบสิกจะนำโปรแกรมไปแจกจ่ายให้สกุลต่าง ๆ พร้อมที่จะรัน

หน้าต่าง Code Window

มีเรื่องประหลาดของวิชวลเบสิกอีกเรื่องหนึ่งคือ การเก็บโปรแกรมเหมือนกับเก็บเป็นหน้า หน้าละ 1 กิจกรรม คุณจึงไม่สามารถเห็นโปรแกรมทั้งหมดครับ

อย่างมากจะเห็นได้ 2 โพรซีเดอร์พร้อมกันเท่านั้น เพราะสามารถแบ่งหน้าต่างนี้ออกเป็น 2 บานได้คือ บานบนกับบานล่าง ถ้าเลื่อนเม้าส์ไปใต้กรอบทั้งสอง เช่นใต้ Proc เคอร์เซอร์จะเปลี่ยนเป็นรูปลูกศรสองทิศทาง ให้คลิ๊กเม้าส์เลื่อนลงล่างแล้วปล่อยมือ

หากต้องการเห็นโปรแกรมทั้งหมดจะต้องเก็บเป็นเท็กซ์ไฟล์โดยเลือกรายการ File/SaveText และถ้าต้องการเก็บโปรแกรมนี้ (อย่างเช่นที่เก็บไฟล์ .pas) ให้เลือกรายการ File/Save Project As จะถามว่าจะเก็บไฟล์ .frm หรือไม่ ให้ตอบ Yes เพราะต้องเก็บไฟล์ 2 ไฟล์ อีกไฟล์หนึ่งคือ .mke ชื่อตามแต่เราจะตั้ง และถ้าต้องการไฟล์ .exe ให้เลือกรายการ File/Make EXE File

คอนสตรัคเตอร์อยู่ไหน

กลับมาเรื่องโอโอพีต่อครับ เมื่อเป็นโอโอพีก็ต้องมีคอนสตรัคเตอร์และดีสตรัคเตอร์ ดังเช่นในเทอร์โบปาสคาลมี Init กับ Done เพื่อให้กำเนิดออปเจ็กต์ด้วย Init และยุบด้วย Done

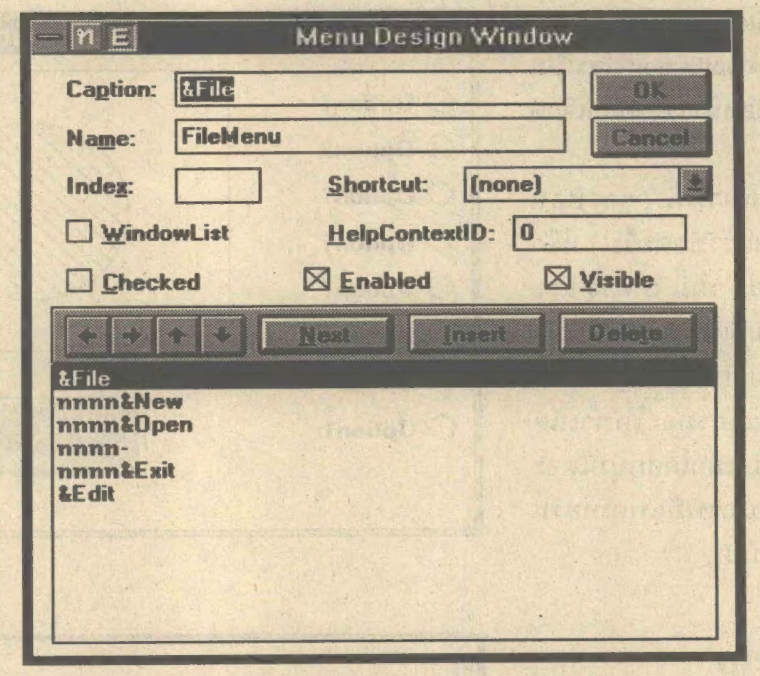
และเราได้อาศัย Init ในการเตรียมค่ารวมทั้งการจองหน่วยความจำ และ Done เพื่อเก็บสิ่งต่าง ๆ ลงไฟล์และยกเลิกหน่วยความจำ

โดยที่เมื่อเราให้กำเนิดออปเจ็กต์แล้วเราจะต้องเป็นผู้ยุบ แต่สำหรับวินโดวส์หากออปเจ็กต์นั้นเป็นหน้าต่างวินโดวส์จะเป็นผู้ยุบเมื่อผู้ใช้ปิดหน้าต่างและจะยุบหน้าต่างลูกทุกบานด้วย เว้นเราต้องการยุบหน้าต่างนั้นก่อน

หากดูใน Command1_Click ในลิสติงที่ 2 จะเห็นว่ามีคำว่า Load ซึ่งในวิชวลเบสิกเรียกเป็น statement ทำหน้าที่เป็นคอนสตรัคเตอร์แต่ไม่ใช่ครับ หากติดตามต่อไปจะพบว่ามี Load อีกคำหนึ่งเป็นกิจกรรมในวิชวลเบสิกเรียกว่า event ไม่ใช่คอนสตรัคเตอร์เช่นกัน

แล้วคอนสตรัคเตอร์ตัวจริงอยู่ที่ไหน ไม่เห็นหรอกครับ ไมโครซอฟต์ซ่อนไว้มิดชิด (กลัวคนกลัวโอโอพีเห็น) โดยให้ในคอนสตรัคเตอร์เรียกใช้ Load ที่เป็น event ให้เราเตรียมค่าต่าง ๆ และสร้าง Load ที่เป็น statement เรียกใช้คอนสตรัคเตอร์อีกทอดหนึ่ง

ทางด้านดีสตรัคเตอร์ก็มี UnLoad ทั้งสอง (ผมไม่ชอบคำ 2 คำนี้ ดูไม่ให้ความหมายที่ตรงตัว)



รูปที่ 7
ไดอะล็อกบ็อกซ์สำหรับสร้างเมนู

เมนู

เพื่อให้ครบเรื่องผมจึงขอนำเรื่องการสร้างเมนูมากล่าวถึง เพราะในวิชวลเบสิกให้รายการในเมนูเป็นออปเจ็กต์ ไม่มีปัญหาหรอกครับเพราะเมนูก็เป็นหน้าต่างเช่นกัน

สมมติว่าจะสร้างเมนูให้มีรายการหลักเป็น File และ Edit มีรายการรองตามมาตรฐาน จะดำเนินการได้ดังนี้คือ

1. เลือกรายการ Window/Menu Design จะแสดงไดอะล็อกบ็อกซ์ให้สร้างเมนูดังในรูปที่ 7 แต่กรอบต่าง ๆ ว่าง

2. ป้อนชื่อรายการเป็น &File ที่กรอบ Caption (เช่นเดียวกับคอนโทรล) ป้อนชื่อออปเจ็กต์ตามแต่จะตั้ง เช่น

FileMenu ที่กรอบ Name (เช่นกัน) แล้วกดคีย์ Enter ที่กรอบ Name เพื่อบ่งว่าจะป้อนรายการต่อไป (รายละเอียดดูได้จากหัวข้อ Menu Design Window)

3. เลือกปุ่มลูกศรขวา เพื่อป้อนรายการรองคือ

- ป้อนชื่อรายการรอง เริ่มจาก New ป้อนชื่อออปเจ็กต์ เช่น อาจให้เป็น cmNew หรือ itmNew หรือจะตั้งอย่างไรก็ได้ แล้วกดคีย์ Enter

- ป้อนรายการรองอื่น ๆ ต่อไป หากต้องการเส้นคั่น ให้ป้อน - (hyphen) เป็นชื่อรายการ และต้องป้อนชื่อออปเจ็กต์ด้วย เช่น อาจให้เป็น cmSepar1

- ป้อนจนถึง Exit

4. เลือกปุ่มลูกศรซ้าย เพื่อป้อนรายการหลัก แล้วป้อนของ Edit จนครบ

5. เลือกปุ่ม OK กลับไปที่ฟอร์มจะแสดงเมนูเหมือนเมื่อขณะรัน

6. ป้อนโปรแกรมให้แต่ละรายการ โดยเลือกรายการนั้น จะแสดงโครงโปรแกรมเช่นเดียวกับของคอนโทรล

เป็นโปรแกรมสนองตอบเมสเสจเช่นกัน หากเทียบกับเทอร์โบปาสคาลก็คือ การรับเมสเสจด้วยฐาน cm_First

ข้างในเป็นโอโอพี

ผมกล่าวมายาวเหยียดก็เพื่อที่จะกล่าวว่า ในวิชวลเบสิกได้จัดโปรแกรมสนองตอบเมสเสจทุกชนิดไว้ในสิ่งที่ควรเรียกว่าเป็นสกุลตามที่สมมติไว้ และตรงนี้ต้องกล่าวว่ามันคือสกุลครับ

ไม่ให้สืบสกุล

คงเหลือแต่เพียงการสืบสกุลที่ให้สืบได้เพียงชั้นเดียว โดยให้เราเขียนโปรแกรมสืบสกุลที่ได้เตรียมไว้แล้ว

และมีเรื่องที่ไม่น่าให้เป็นคือ ฟอร์มตามที่ผมกล่าวมาในวิชวลเบสิกเรียกว่า Class ก็คือสกุล สิ่งใดก็ตาม (แบบข้อมูล ตัวแปร ค่าคงตัว โปรแกรมย่อย และกิจกรรม) ที่กำหนดไว้ในฟอร์มหนึ่งจะเป็นของฟอร์มนั้นเท่านั้น ฟอร์มอื่นจะไม่รู้จัก เป็นการปกป้อง เป็น encapsulation

แต่ก็ให้ฟอร์มอื่นเข้าถึงได้บ้างในระดับเรคอร์ด (เช่นเดียวกับในเทอร์โบปาสคาล) แต่เฉพาะฟิลด์ที่เป็น Properties กับกิจกรรมที่ในวิชวลเบสิกเรียกว่า Method เช่น เราอาจเขียนโปรแกรมว่า

```
Form1.Text1.SetFocus
```

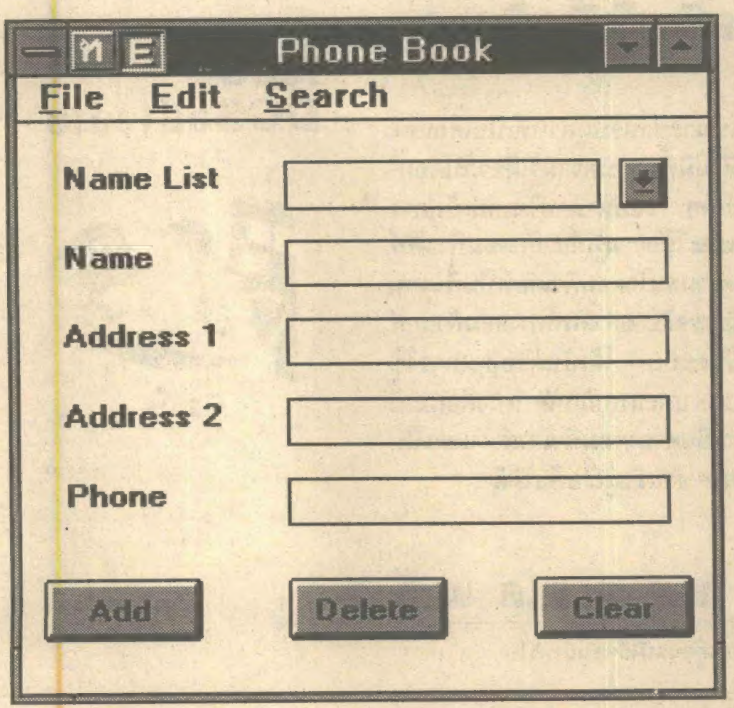
เช่นเขียนไว้ในฟอร์ม Form2. Command1 ที่น่าประหลาดคือ ทำไมถึงยอมให้สองฟิลด์นี้เท่านั้น หากให้เข้าถึงเรื่องอื่น ๆ ด้วยก็จะได้ว่าฟอร์มแต่ละฟอร์มคือสกุลหนึ่ง ๆ และเขียนโปรแกรมสืบสกุลกันได้ เช่น ให้ Form2 สืบสกุลจาก Form1 ก็จะเป็นโอโอพี ผมไม่เห็นข้อขัดข้องอันใด และไม่เห็นเหตุผลด้วยนอกจากกลัวคนกลัวโอโอพี

ทำให้เป็นสกุล

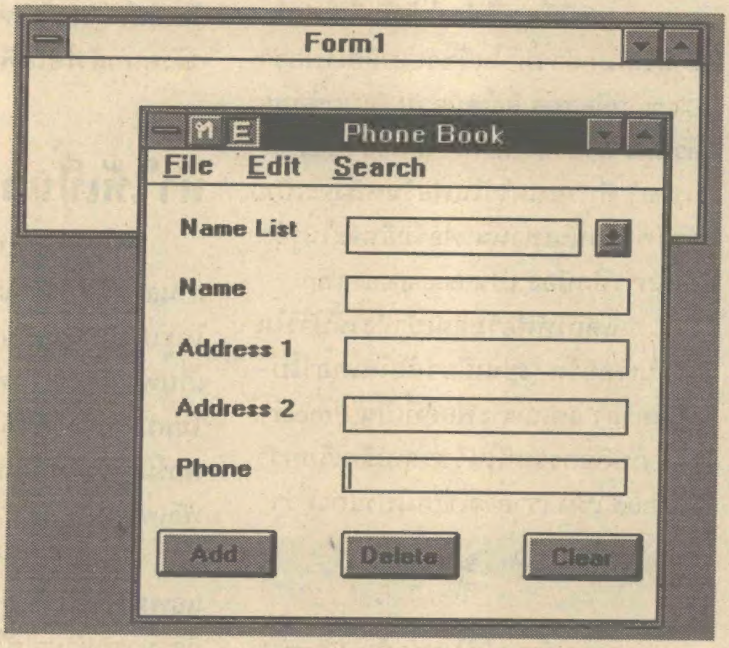
ทั้งรู้ ๆ ครับว่าเขาไม่ต้องการให้ทำเป็นสกุลก็พยายามทำให้เป็นสกุล ขอให้ดูในรูปที่ 8 ผมเขียนโปรแกรมไว้สำหรับเก็บหมายเลขโทรศัพท์ (โปรแกรมจริงยังไม่สมบูรณ์) จะเห็นว่าผมใช้ฟอร์มเพียงฟอร์มเดียว จึงเป็นหน้าต่างหลักของแอพพลิเคชันนี้

ต่อมาผมเปิดโปรเจ็กต์ใหม่เป็นแอพพลิเคชันใหม่ต้องการสมุดโทรศัพท์ก็สามารถนำมาใช้ได้ทันทีโดยเปิดฟอร์มที่เป็นสมุดโทรศัพท์ด้วยรายการ File/Add File จะได้สมุดโทรศัพท์ดังรูปที่ 9 มาใช้ในแอพพลิเคชันใหม่นี้

ดังนั้นเจ้าฟอร์มที่เป็นสมุดโทรศัพท์คือสกุลในสายตาของผม และในวิชวลเบสิกก็บอกไปไว้ด้วยโดยเรียกชื่อสกุล



รูปที่ 8
หน้าต่างของโปรแกรม PhoneBook



รูปที่ 9
แสดงการใช้ฟอร์มซ้ำในโปรแกรมอื่น

(Name) เป็น Class

ซึ่งเป็นวัตถุประสงค์ของการเขีย
โปรแกรมในแบบโอโอพีที่ ให้สามารถน
โปรแกรมเก่า (ที่เขียนในวันนี้) ไปใช้ไ
ใหม่ (ในวันหน้า)

ท้ายตอน

ผมจึงถามเขาบ้างว่า เห็นหรือยั
ว่าเป็นโอโอพี เขาไม่ตอบ จึงไม่ทราบว่า
ผมทำบาปหรือเปล่า หากเขาเกิดกลัวเง
โอโอพี

เรียน
DOSอย่างง่าย
สำหรับผู้เริ่มต้น

เหมาะสำหรับผู้อ่านที่เป็นเด็กและผู้ใหญ่ที่ไม่มีพื้นฐานความรู้เกี่ยวกับคอมพิวเตอร์เลย เนื้อหาภายในเล่มเน้นการส่งเสริมและให้ความรู้เกี่ยวกับคอมพิวเตอร์แก่ผู้ที่มีความสนใจคอมพิวเตอร์เป็นพื้นฐาน รวมทั้งมีการอธิบายคำศัพท์ทางคอมพิวเตอร์อย่างเป็นระบบ ได้เน้นตัวอย่างการใช้คำสั่ง DOS และแบบฝึกหัด หนังสือเล่มนี้จึงนับว่าเป็นตำราคอมพิวเตอร์ และเป็นคู่มือ DOS สำหรับผู้เริ่มต้นได้ดี

เรียน
DOSอย่างง่าย

ดร.วิทยา
เรืองพรวิสุทธิ์

วางตลาดแล้ว

หาซื้อได้ตามร้านหนังสือชั้นนำทั่วไป